
Subject: Re: User lists of "bad" naming of classes, functions etc in U++...

Posted by [andrei_natanael](#) on Tue, 17 Nov 2009 14:41:17 GMT

[View Forum Message](#) <> [Reply to Message](#)

ag_newb wrote on Tue, 17 November 2009 11:00 Boost C++ naming system should be adopted.
reason: delimiting by underscore is safer than capitalization, IMHO.
Have you an argument for this?

```
void DoSomething(const String& paramOne) {}
```

```
void do_something(const string& param_one) {}
```

IMO, using second version(with underscores) takes more time to write because when you write _ you have to press Shift+_ and that means 2 key press more than camel case naming.

Using camelCase notation you identify clearly which is a method of a class or a public variable. For example, you could notate all member function using notation LikeThis and variables using notation likeThis. If a variable is an instance of a class which redefine operator () is clear that accessing it like instance.variableName() means calling operator () from variableName and not calling function variableName().

```
class X
{
    class Y { public: void operator() {} };
public:
    Y variableName;
    void CallMe() {}
};
```

```
class x
{
    class y { public: void operator() {} };
public:
    y variable_name;
    void call_me() {}
};
```

```
int main()
{
    X camelCaseNotation;
    x underscore_notation;
    camelCaseNotation.variableName(); // means calling operator () from variableName
    camelCaseNotation.CallMe(); // means calling a member function
```

```
    underscore_notation.variable_name(); // looks like we are calling a member function
    underscore_notation.call_me(); // ok, we are calling a member function
    return 0;
```

```
}
```

IMO camelCase make you easier understanding the context.
